Capstone_Berlin_Stations_organicFood

June 7, 2021

++ This notebook will mainly be used for the Coursera Capstone Project +++

1 Capstone Project Coursera for Data Science: The battle of neighborhoods or the battle of "organic Food & Beverages" at the close vicinity of Berlin metro stations

1.0.1 IBM Data Science Course, attendee Dr.B.Bayer, June 2021

1.1 Introduction

1.1.1 Background Information

Berlin is the capital of Germany with about 3.7 million inhabitants and a city full of movement. On average, every Berliner travels more than three times a day. Local public transport plays a special role in this. Approximately 50 percent of Berlin's households are car-free, and with 324 cars per 1,000 inhabitants, the city has the lowest motorization rate in Germany. In addition, even people who do have a car often use other means of transportation. Public transportation also plays a central role for commuting to work. For example, about 40% of all commuters use public transportation. [1]

Another strongly growing trend is the demand for organic products and therefore incoming, the desire for a healthy diet [2]. In train stations and metro stations, there are already numerous possibilities for food intake, and so numerous small and large stores offer food and drinks to take away. However, these offers are very often not considered healthy at all and thus contradict the desire for a healthy diet. The market for healthy take-away products is still relatively small and this results in a large potential market for investors.

1.1.2 Problem Statement

With the aforementioned prospect, various stakeholders (entrepreneurs, investors) may be interested to explore the organic food and beverage (F&B) shop business opportunities in the very close vicinity of Berlins metro stations. This data science project is thus carried out to help them answer the following question: Which of the Berlins metro stations are strategic for opening an "organic F&B" business?

1.2 Data

In order to explore potential answer to the problem, the following data are required:

- Metro stations of Berlin city with their geographical coordinates [3]. They are required to utilize Foursquare API in the subsequent step.
- Information about venues of the station: the names, category, venue latitudes, venue longitudes. These are obtained using Foursquare API [4]. The stations will be clustered based on their venues to find the best location candidates for "organic F&B".

1.3 Methodology

This section represents the main components of the report. It starts with data extraction (web scraping) of Berlin metro stations and retrieval of geographical coordinates. Leveraging Foursquare API, these coordinates data are given as inputs to explore venues within the stations.

One-hot encoding is performed to analyze and narrow down the most common venues in each of the station. Given all the venues surrounding them, the stations are clustered using K-means algorithm. The number of optimal clusters is decided using the elbow method and silhouette score. Each cluster is separately analyzed to examine one discriminating venue that characterizes them. Analysis of the clusters and visualization will give insights as to where the strategic regions to set up the business.

The following cell contains all the necessary Python libraries.

[1]:
```
# import libraries
import numpy as np
import pandas as pd
import requests
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import json, lxml

from pandas.io.json import json_normalize # tranform JSON file into a pandas␣
↪dataframe
#from geopy.geocoders import Nominatim # convert an address into latitude and␣
↪longitude values

from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from bs4 import BeautifulSoup

try:
    import folium
except:
    !pip install folium
    import folium
```

```
try:
    import geopy
except:
    !pip install geopy
    import geopy
from geopy.geocoders import Nominatim

print("Libraries installed.")
```

```
Libraries installed.
```

Two main dataframes will be created for use in the analysis:

- data_stat: contains names and geographical coordinates of all Berlin metro (tram) stations.
- station_venues_all: contains at most 100 venues and venues details (name, category, latitude, longitude) for every metro stations.

1.4 Web Scraping: Berlin tram stations and coordinates

The data to scrape are the names of all Berlin tram stations and their corresponding geographical coordinates. We first need to specify all the URLs of the webpages to which we will send a get request. For reference, Berlins tram stations are listed on Wikipedia page:

- https://de.wikipedia.org/wiki/Liste_der_Berliner_U-Bahnh%C3%B6fe

[9]:
```
# Read the data of boroughs and localities of great Berlin from Wikipedia
df = pd.read_html('https://de.wikipedia.org/wiki/
↪Liste_der_Berliner_U-Bahnh%C3%B6fe')
```

[10]:
```
data = df[1]
```

[11]:
```
data
```

[11]:
```
                             Bahnhof (Kürzel) Karte  Linie      Eröffnung  \
0     Adenauerplatz (Ado) 52° 29  59  N, 13° 18  26  O    NaN  28. Apr. 1978
1    Afrikanische Straße (Afr) 52° 33  38  N, 13° 2…    NaN    3. Mai 1956
2      Alexanderplatz (A) 52° 31  17  N, 13° 24  48  O    NaN   1. Juli 1913
3     Alexanderplatz (Al) 52° 31  17  N, 13° 24  48  O    NaN  21. Dez. 1930
4     Alexanderplatz (Ap) 52° 31  17  N, 13° 24  48  O    NaN  18. Apr. 1930
..                                                  …      …              …
195       Yorckstraße (Y) 52° 29  35  N, 13° 22  15  O    NaN  29. Jan. 1971
196         Zitadelle (Zi) 52° 32  16  N, 13° 13  4  O    NaN   1. Okt. 1984
197  Zoologischer Garten (Zo) 52° 30  26  N, 13° 19…    NaN  11. März 1902
198  Zoologischer Garten (Zu) 52° 30  26  N, 13° 19…    NaN  28. Aug. 1961
199    Zwickauer Damm (Zd) 52° 25  24  N, 13° 29  2  O    NaN   2. Jan. 1970

       Lage        Ortsteil  Umstieg  \
0    Tunnel  Charlottenburg      NaN
1    Tunnel         Wedding      NaN
2    Tunnel           Mitte      NaN
```

```
3    Tunnel           Mitte      NaN
4    Tunnel           Mitte      NaN
..      …               …       …
195  Tunnel      Schöneberg      NaN
196  Tunnel      Haselhorst      NaN
197  Tunnel  Charlottenburg      NaN
198  Tunnel  Charlottenburg      NaN
199  Tunnel    Gropiusstadt      NaN

                                    Denkmal                 Anmerkungen  \
0                                         -                         NaN
1                                         -                         NaN
2    Eintrag in der Berliner Landesdenkmalliste                     NaN
3    Eintrag in der Berliner Landesdenkmalliste                     NaN
4    Eintrag in der Berliner Landesdenkmalliste  1961-1990 „Geisterbahnhof"
..                                        …                          …
195                                       -                         NaN
196  Eintrag in der Berliner Landesdenkmalliste                     NaN
197  Eintrag in der Berliner Landesdenkmalliste                     NaN
198  Eintrag in der Berliner Landesdenkmalliste                     NaN
199  Eintrag in der Berliner Landesdenkmalliste                     NaN

                                Sehenswürdigkeiten  Bild
0                                              NaN   NaN
1                                              NaN   NaN
2    Alexa, Fernsehturm, Haus des Lehrers, Urania-W…   NaN
3    Alexa, Fernsehturm, Haus des Lehrers, Urania-W…   NaN
4    Alexa, Fernsehturm, Haus des Lehrers, Urania-W…   NaN
..                                               …    …
195                 Yorckbrücken, St.-Matthäus-Kirchhof   NaN
196                                  Zitadelle Spandau   NaN
197  Zoologischer Garten, Schillertheater, Kaiser-W…   NaN
198  Zoologischer Garten, Schillertheater, Kaiser-W…   NaN
199                                              NaN   NaN

[200 rows x 10 columns]
```

1.4.1 Data Cleaning

Keep only columns Station name (incl. coordinates) and locality

[12]:
```
data.
↪drop(["Linie","Eröffnung","Lage","Umstieg","Denkmal","Anmerkungen","Sehenswürdigkeiten","Bi
↪axis=1, inplace=True)
```

[13]:
```
data.rename(columns = {"Bahnhof (Kürzel) Karte" : "Stationname", "Ortsteil" :␣
↪"Locality"}, inplace = True)
```

[14]:
```
data
```

[14]:
```
                                          Stationname        Locality
0     Adenauerplatz (Ado) 52° 29  59  N, 13° 18  26  O  Charlottenburg
1    Afrikanische Straße (Afr) 52° 33  38  N, 13° 2…          Wedding
2      Alexanderplatz (A) 52° 31  17  N, 13° 24  48  O            Mitte
3     Alexanderplatz (Al) 52° 31  17  N, 13° 24  48  O            Mitte
4     Alexanderplatz (Ap) 52° 31  17  N, 13° 24  48  O            Mitte
..                                                 …              …
195       Yorckstraße (Y) 52° 29  35  N, 13° 22  15  O      Schöneberg
196         Zitadelle (Zi) 52° 32  16  N, 13° 13  4  O      Haselhorst
197  Zoologischer Garten (Zo) 52° 30  26  N, 13° 19…  Charlottenburg
198  Zoologischer Garten (Zu) 52° 30  26  N, 13° 19…  Charlottenburg
199    Zwickauer Damm (Zd) 52° 25  24  N, 13° 29  2  O    Gropiusstadt

[200 rows x 2 columns]
```

Some more cleaning

[15]:
```
# manual data cleaning after inspection of the downloaded data
# data.to_excel("output.xlsx")
data['Stationname'][109] = 'Museumsinsel (MU) 52° 31  3  N, 13° 23  54  O'
data['Stationname'][148] = 'Rotes Rathaus (RR) 52° 31  7  N, 13° 24  30  O'
data['Stationname'][178] = 'Unter den Linden (Uli) 52° 31  1  N, 13° 23  20  O'
data['Stationname'][179] = 'Unter den Linden (Uli) 52° 31  1  N, 13° 23  20  O'
```

[16]:
```
# Data cleaning to get station name and latidude/Longitude
# Stationsname
a = data["Stationname"].str.split("(", n=1, expand = True)
data["Station"] = a[0]
```

[17]:
```
# LatLong preparation as str --> final convert to decimal
b = data["Stationname"].str.rsplit(")", n=1, expand = True)
data["Koord"] = b[1]
# SPLIT INTO TWO STRINGS
c = data["Koord"].str.split("N,", n=1, expand = True)
data["Lat"] = c[0]
data["Long"] = c[1]
data["Long"] = data["Long"].str[:-1]
d = data["Long"].str.split("°", n=1, expand = True)
long_grad = d[0]
e = d[1].str.split(" ", n=1, expand = True)
long_min = e[0]
long_sec = e[1].str[:-2]
dd = data["Lat"].str.split("°", n=1, expand = True)
lat_grad = dd[0]
ee = dd[1].str.split(" ", n=1, expand = True)
lat_min = ee[0]
```

```
lat_sec = ee[1].str[:-2]  # remove leerzeichen and sec character
data["lat_grad"] = lat_grad.str.strip()
data["lat_min"] = lat_min.str.strip()
data["lat_sec"] = lat_sec.str.strip()
data["long_grad"] = long_grad.str.strip()
data["long_min"] = long_min.str.strip()
data["long_sec"] = long_sec.str.strip()
```

[18]:

```
data.head(100)
```

[18]:

```
                                        Stationname        Locality  \
0    Adenauerplatz (Ado) 52° 29  59  N, 13° 18  26  O  Charlottenburg
1   Afrikanische Straße (Afr) 52° 33  38  N, 13° 2…         Wedding
2     Alexanderplatz (A) 52° 31  17  N, 13° 24  48  O           Mitte
3    Alexanderplatz (Al) 52° 31  17  N, 13° 24  48  O           Mitte
4    Alexanderplatz (Ap) 52° 31  17  N, 13° 24  48  O           Mitte
..                                              …               …
95    Leopoldplatz (Lpu) 52° 32  47  N, 13° 21  33  O         Wedding
96      Lichtenberg (Li) 52° 30  38  N, 13° 29  47  O     Lichtenberg
97   Lindauer Allee (LD) 52° 34  31  N, 13° 20  21  O   Reinickendorf
98   Lipschitzallee (La) 52° 25  29  N, 13° 27  46  O    Gropiusstadt
99  Louis-Lewin-Straße (LL) 52° 32  20  N, 13° 37 …     Hellersdorf

                 Station                          Koord            Lat  \
0          Adenauerplatz    52° 29  59  N, 13° 18  26  O    52° 29  59
1    Afrikanische Straße     52° 33  38  N, 13° 20  3  O    52° 33  38
2         Alexanderplatz    52° 31  17  N, 13° 24  48  O    52° 31  17
3         Alexanderplatz    52° 31  17  N, 13° 24  48  O    52° 31  17
4         Alexanderplatz    52° 31  17  N, 13° 24  48  O    52° 31  17
..                   …                              …             …
95          Leopoldplatz    52° 32  47  N, 13° 21  33  O    52° 32  47
96           Lichtenberg    52° 30  38  N, 13° 29  47  O    52° 30  38
97        Lindauer Allee    52° 34  31  N, 13° 20  21  O    52° 34  31
98        Lipschitzallee    52° 25  29  N, 13° 27  46  O    52° 25  29
99    Louis-Lewin-Straße     52° 32  20  N, 13° 37  6  O    52° 32  20

             Long lat_grad lat_min lat_sec long_grad long_min long_sec
0    13° 18  26         52      29      59        13       18       26
1     13° 20  3         52      33      38        13       20        3
2    13° 24  48         52      31      17        13       24       48
3    13° 24  48         52      31      17        13       24       48
4    13° 24  48         52      31      17        13       24       48
..            …      …     …     …       …      …      …
95   13° 21  33         52      32      47        13       21       33
96   13° 29  47         52      30      38        13       29       47
97   13° 20  21         52      34      31        13       20       21
98   13° 27  46         52      25      29        13       27       46
```

```
99    13° 37  6         52      32      20        13       37         6

[100 rows x 12 columns]
```

[19]:
```
#data.to_excel("output.xlsx")
```

[20]:
```
data["lat_grad"] = data["lat_grad"].astype(str).astype(float, errors = 'raise')
data["lat_min"]  = data["lat_min"].astype(str).astype(float, errors = 'raise')
data["lat_sec"]  = data["lat_sec"].astype(str).astype(float, errors = 'raise')
data["long_grad"]= data["long_grad"].astype(str).astype(float, errors = 'raise')
data["long_min"] = data["long_min"].astype(str).astype(float, errors = 'raise')
data["long_sec"] = data["long_sec"].astype(str).astype(float, errors = 'raise')
```

[21]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Stationname  200 non-null    object
 1   Locality     200 non-null    object
 2   Station      200 non-null    object
 3   Koord        200 non-null    object
 4   Lat          200 non-null    object
 5   Long         200 non-null    object
 6   lat_grad     200 non-null    float64
 7   lat_min      200 non-null    float64
 8   lat_sec      200 non-null    float64
 9   long_grad    200 non-null    float64
 10  long_min     200 non-null    float64
 11  long_sec     200 non-null    float64
dtypes: float64(6), object(6)
memory usage: 18.9+ KB
```

Geograhical coordinates conversion

[22]:
```
# Convert coordinates given in degree to dezimal
data['Latitude'] = ((( data['lat_sec'] / 60 ) +  data['lat_min']) / 60) +␣
↪data['lat_grad']
data['Longitude'] = ((( data['long_sec'] / 60 ) +  data['long_min']) / 60) +␣
↪data['long_grad']
```

[23]:
```
data['Latitude']
```

[23]:
```
0      52.499722
1      52.560556
2      52.521389
```

```
3      52.521389
4      52.521389
         …
195    52.493056
196    52.537778
197    52.507222
198    52.507222
199    52.423333
Name: Latitude, Length: 200, dtype: float64
```

[24]:
```
# Prepare new dataframe
data.
↪drop(["Stationname","Koord","Lat","Long","lat_grad","lat_min","lat_sec","long_grad","long_m
↪axis=1, inplace=True)
```

[25]:
```
# last but not least remove double entries for stations
# df_without_duplicates = df_with_duplicates.drop_duplicates(subset=['Name'])
data_stat = data.drop_duplicates(subset=['Station'])
```

[26]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Locality   200 non-null    object
 1   Station    200 non-null    object
 2   Latitude   200 non-null    float64
 3   Longitude  200 non-null    float64
dtypes: float64(2), object(2)
memory usage: 6.4+ KB
```

[27]:
```
data_stat
```

[27]:

	Locality	Station	Latitude	Longitude
0	Charlottenburg	Adenauerplatz	52.499722	13.307222
1	Wedding	Afrikanische Straße	52.560556	13.334167
2	Mitte	Alexanderplatz	52.521389	13.413333
5	Spandau	Altstadt Spandau	52.539167	13.205556
6	Mariendorf	Alt-Mariendorf	52.439722	13.387500
..	…	…	…	…
194	Gropiusstadt	Wutzkyallee	52.423333	13.474722
195	Schöneberg	Yorckstraße	52.493056	13.370833
196	Haselhorst	Zitadelle	52.537778	13.217778
197	Charlottenburg	Zoologischer Garten	52.507222	13.332500
199	Gropiusstadt	Zwickauer Damm	52.423333	13.483889

```
[178 rows x 4 columns]
```

[28]:
```python
# dump to excel
#data_stat.to_excel("stations_berlin.xlsx")
```

1.4.2 Plot a nice map to show the stations

A total of 178 tram stations are present and taken into consideration. Note the gap of tram stations in the eastern part of the city. This is due to the former separation of the city. The eastern part is still not very well developed w.r.t public transport via tram.

[29]:
```python
address = 'Berlin'
geolocator = Nominatim(user_agent="Berlin")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinates of Berlin are {}, {}.'.format(latitude,␣
 ↪longitude))
```

```
The geograpical coordinates of Berlin are 52.5170365, 13.3888599.
```

[31]:
```python
# create map of Berlin boroughs using latitude and longitude
map_berlin_stations = folium.Map(location=[latitude, longitude], zoom_start=11)
# add markers to the map
for lat, lng, label in zip(data_stat['Latitude'], data_stat['Longitude'],␣
 ↪data_stat['Station']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker([lat,␣
 ↪lng],radius=5,popup=label,color='blue',fill=True,fill_color='#3186cc',fill_opacity=0.
 ↪7,parse_html=False).add_to(map_berlin_stations)
map_berlin_stations
```

[31]:
```
<folium.folium.Map at 0xdc60d00>
```

1.5 Foursquare venue data

Before exploring the venues using Foursquare API, credentials and version must first be defined. In the publication version the credentials were removed.

[32]:
```python
# Foursquare access data
CLIENT_ID = '_deleted_for_publication_'     # your Foursquare ID
CLIENT_SECRET = '_deleted_for_publication_' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
```

Define a function with search radius of 100 m around the stations coordinates. Venue limit entry 200. The function will return a dataframe containing venues within defined radius of a region (i.e., a subdistrict), with the following details: venue name, venue category, venue latitude, venue longitude. The inputs to be provided are the names of the city, district, subdistrict, as well as the latitudes and longitudes.

```
[33]: # define the latitude and longitude using above created dataframe
lat = data_stat['Latitude'] # stations latitude value
lon = data_stat['Longitude'] # stations longitude value
LIMIT = 200
# radius = 5000 --> Input directly as argument into the function below
```

```
[34]: # Function to get the venues from Foursquare API
def get_near_by_venues(names, latitudes, longitudes, radius=100):
    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?
 ↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'\
        .format(CLIENT_ID, CLIENT_SECRET, VERSION, lat, lng, radius, LIMIT)
        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']
        # return only relevant information for each nearby venue
        venues_list.append([(name, lat, lng,
                             v['venue']['name'], v['venue']['location']['lat'],␣
 ↪v['venue']['location']['lng'],
                             v['venue']['categories'][0]['name']) for v in␣
 ↪results])
    nearby_venues = pd.DataFrame([item for venue in venues_list for item in␣
 ↪venue])
    nearby_venues.columns = ['Station','Station Latitude', 'Station Longitude',
                             'Venue', 'Venue Latitude', 'Venue Longitude',␣
 ↪'Venue Category']
    return nearby_venues
```

Apply the function and save the results in a pandas dataframe (This step may need several minutes.)

```
[36]: station_venues_all =␣
 ↪get_near_by_venues(names=data_stat['Station'],latitudes=data_stat['Latitude'],longitudes=da
```

```
[37]: station_venues_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 842 entries, 0 to 841
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Station            842 non-null    object
 1   Station Latitude   842 non-null    float64
 2   Station Longitude  842 non-null    float64
 3   Venue              842 non-null    object
 4   Venue Latitude     842 non-null    float64
```

```
 5   Venue Longitude    842 non-null    float64
 6   Venue Category     842 non-null    object
dtypes: float64(4), object(3)
memory usage: 46.2+ KB
```

[38]:
```
station_venues_all.head()
```

[38]:
```
          Station  Station Latitude  Station Longitude           Venue  \
0  Adenauerplatz         52.499722          13.307222        Bellucci
1  Adenauerplatz         52.499722          13.307222     Block House
2  Adenauerplatz         52.499722          13.307222   Adenauerplatz
3  Adenauerplatz         52.499722          13.307222  Einstein Kaffee
4  Adenauerplatz         52.499722          13.307222        Rossmann

   Venue Latitude  Venue Longitude      Venue Category
0       52.499430        13.306800  Italian Restaurant
1       52.499645        13.306755          Steakhouse
2       52.499932        13.307214               Plaza
3       52.500056        13.306058         Coffee Shop
4       52.500013        13.308113           Drugstore
```

[39]:
```
# dump the result into an excel file
#station_venues_all.to_excel("venues_search100_berlin.xlsx")
```

[40]:
```
station_venues_all['Venue Category'].unique()
```

[40]:
```
array(['Italian Restaurant', 'Steakhouse', 'Plaza', 'Coffee Shop',
       'Drugstore', 'Bistro', 'Bakery', 'Boarding House', 'Cocktail Bar',
       'Hookah Bar', 'Metro Station', 'Argentinian Restaurant',
       'Burger Joint', 'Doner Restaurant', 'Neighborhood',
       'Movie Theater', 'Greek Restaurant', 'Bus Stop', 'Clothing Store',
       'Cosmetics Shop', 'Fountain', 'Fried Chicken Joint',
       'Vietnamese Restaurant', 'Pharmacy', 'Turkish Restaurant', 'Café',
       'Spanish Restaurant', 'Currywurst Joint', 'German Restaurant',
       'Supermarket', 'Bank', 'Wine Shop', 'Convenience Store',
       'Chinese Restaurant', 'Grocery Store', 'Organic Grocery',
       'Vegetarian / Vegan Restaurant', 'Bar', 'Yoga Studio', 'Park',
       'Breakfast Spot', 'Restaurant', 'Asian Restaurant',
       'Gym / Fitness Center', 'Vacation Rental', 'Sushi Restaurant',
       'Pizza Place', 'Dive Bar', 'Photography Studio', 'Thai Restaurant',
       'Shopping Mall', 'Hotel', 'Spa', 'Museum', 'Hotel Bar',
       'Modern European Restaurant', 'Donut Shop', 'Mexican Restaurant',
       'Dessert Shop', 'Shoe Store', 'Middle Eastern Restaurant',
       'IT Services', 'Taco Place', 'Indian Restaurant',
       'Furniture / Home Store', 'Flea Market', 'Fast Food Restaurant',
       'Kebab Restaurant', "Women's Store", 'Thrift / Vintage Store',
       'Tram Station', 'Gourmet Shop', 'Chocolate Shop',
       'Department Store', "Men's Store", 'Historic Site', 'Roof Deck',
```

```
'History Museum', 'Kumpir Restaurant', 'Falafel Restaurant',
'African Restaurant', 'Art Gallery', 'Miscellaneous Shop',
'Persian Restaurant', 'BBQ Joint', 'Farmers Market',
'Sandwich Place', 'Train Station', 'Seafood Restaurant',
'Juice Bar', 'Salad Place', 'Newsstand', 'Ice Cream Shop',
'Electronics Store', 'Fish & Chips Shop', 'Smoke Shop', 'Platform',
'Business Service', 'Light Rail Station', 'Pet Store', 'Creperie',
'Mobile Phone Shop', 'Lounge', 'Ukrainian Restaurant',
'Soccer Field', 'Beach Bar', 'Event Space',
'Indonesian Restaurant', 'Taverna', 'Japanese Restaurant',
'Escape Room', 'Tapas Restaurant', 'Playground', 'Gym',
'Music Store', 'Kurdish Restaurant', 'Nightclub', 'Hostel',
'Bosnian Restaurant', 'Sports Bar', 'Comedy Club', 'Pub',
'Food & Drink Shop', 'Arts & Crafts Store', 'Garden', 'Bookstore',
'Korean Restaurant', 'Brasserie', 'Hot Dog Joint',
'Israeli Restaurant', 'Bridge', 'Schnitzel Restaurant',
'Halal Restaurant', 'ATM', 'Brazilian Restaurant', 'Laundromat',
'Scenic Lookout', 'Frozen Yogurt Shop', 'Salon / Barbershop',
'Pool', 'Cigkofte Place', 'Big Box Store', 'Concert Hall',
'Public Bathroom', 'Beer Bar', 'Camera Store', 'French Restaurant',
'Record Shop', 'Costume Shop', 'Wine Bar',
'Paper / Office Supplies Store', 'Indie Movie Theater',
'Outdoor Supply Store', 'Bike Shop', 'Silesian Restaurant',
'Music Venue', 'Lebanese Restaurant', 'Beer Store',
'Sporting Goods Shop', 'Toy / Game Store', 'Post Office',
'Gay Bar', 'Taiwanese Restaurant', 'Mediterranean Restaurant',
'Food', 'General Entertainment', 'Snack Place', 'Adult Boutique',
'Trattoria/Osteria', 'Eastern European Restaurant',
'Deli / Bodega', 'Liquor Store', 'Perfume Shop', 'Exhibit',
'Souvenir Shop', 'Russian Restaurant', 'Theater', 'Gastropub',
'Karaoke Bar', 'Memorial Site', 'Medical Center', 'Burrito Place',
'Carpet Store', 'Art Museum'], dtype=object)
```

Removal of some unwanted categories such as “Metro Station” or “bus stop”. Mainly categories which have nothing in common with food supply.

[41]:
```
removal_list = ['Drugstore', 'Bus Stop', 'Metro Station', 'Gift Shop',␣
↪'Clothing Store', 'Bookstore', 'Cosmetics Shop',
      'Department Store', 'Electronics Store', 'Shoe Store','Neighborhood',␣
↪'Mobile Phone Shop', 'Movie Theater','Supermarket',
      'Arts & Crafts Store', 'Liquor Store','Paper / Office Supplies Store',␣
↪'Bank', 'Grocery Store', 'Gym / Fitness Center', 'Flower Shop',
      'Organic Grocery', 'Vacation Rental', 'Athletics & Sports', 'Indie Movie␣
↪Theater', 'Spa', 'Art Gallery', 'Museum', 'Pharmacy',
      'Big Box Store', 'Park', 'Farm', 'Garden Center', 'Gym','Opera House',␣
↪'Sporting Goods Shop', 'IT Services',
```

```
      'Furniture / Home Store', 'Hobby Shop', 'Roof Deck','Health & Beauty␣
↪Service', 'History Museum','Photography Studio', 'Playground',
      'Pet Store', 'Theater', 'Public Bathroom','Flea Market', 'Toy / Game␣
↪Store', 'Thrift / Vintage Store', 'Tram Station',
      'Mini Golf', 'Chocolate Shop', 'Optical Shop', 'Boutique',"Men's Store",␣
↪'Lake', 'Shipping Store',
      'Historic Site', 'Music Venue', 'Medical Center', 'Camera Store''Science␣
↪Museum', 'Souvenir Shop', 'ATM',
      'Record Shop', 'Convenience Store','Automotive Shop', 'Miscellaneous␣
↪Shop', 'Boarding House',
      'Cafeteria', 'Farmers Market', 'Hotel','Train Station', 'Salad Place',␣
↪'Newsstand','Smoke Shop', 'Taxi Stand', 'Lingerie Store',
      'Hot Spring', 'Nightclub', 'Beer Garden', 'Hostel','Yoga Studio', 'Light␣
↪Rail Station', 'Performing Arts Venue',
      'Soccer Field', 'Beach Bar', 'Event Space', 'Multiplex','Platform',␣
↪'Post Office', 'Indie Theater', 'Discount Store',
      'Luggage Store', 'Escape Room', 'Coworking Space','Gas Station', 'Music␣
↪Store', 'Lounge','Monument / Landmark', 'Plaza',
      "Women's Store", 'Speakeasy', 'Adult Boutique','Rental Car␣
↪Location','Campground','Dance Studio', 'Comedy Club', 'Massage Studio',
      'Other Repair Shop', 'Bath House', 'Deli / Bodega',␣
↪'Waterfront','Garden','Trail','Martial Arts School',
      'Gay Bar', 'Hot Dog Joint','Circus', 'Gym Pool', 'Bridge', 'Rock Club',␣
↪'Stationery Store',
      'Pool', 'Business Service', 'Shopping Plaza','Hardware Store',␣
↪'Laundromat', 'Building','Scenic Lookout', 'Country Dance Club',
      'Concert Hall', 'Fountain', 'Wings Joint', 'Sculpture Garden', 'Plaza',␣
↪'Camera Store', 'Bubble Tea Shop',
      'Costume Shop',  'Video Store','Outdoor Supply Store', 'Bike Shop',␣
↪'General Entertainment', 'Art Museum', 'Sauna / Steam Room',
      'Bridal Shop','Print Shop', 'Intersection', 'Zoo Exhibit','Harbor /␣
↪Marina', 'Pool Hall', 'Exhibit', 'Perfume Shop', 'Karaoke Bar',
       'Salon / Barbershop','Memorial Site', 'Jewelry Store', 'Carpet Store',␣
↪'Bike Rental / Bike Share']
```

[42]:
```
station_venues_toeat = station_venues_all[~station_venues_all['Venue Category'].
↪isin(removal_list)]
```

[43]:
```
print('Uniques categories for all stations of Berlin: {}'.
↪format(len(station_venues_all['Venue Category'].unique())))
print('Uniques categories after modification: {}'.
↪format(len(station_venues_toeat['Venue Category'].unique())))
```

```
Uniques categories for all stations of Berlin: 184
Uniques categories after modification: 85
```

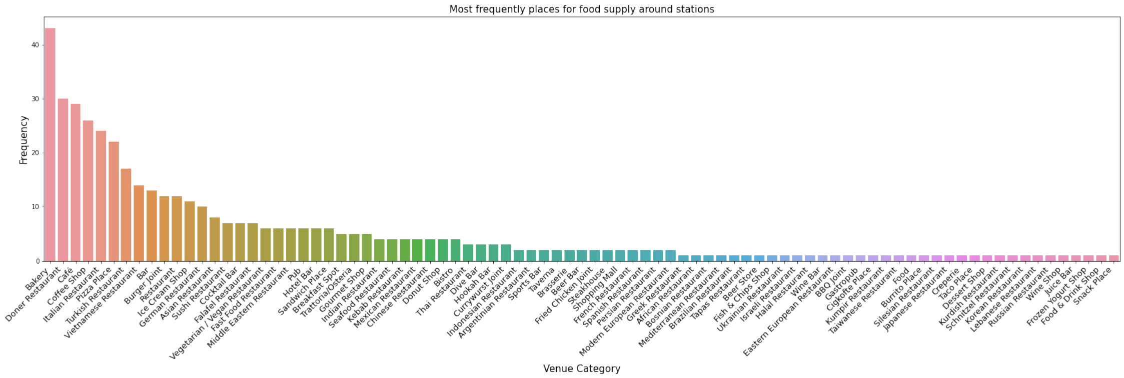
Plot for eating places

```
[44]: # create a dataframe of top 100 categories
Berlin_places2eat = station_venues_toeat['Venue Category'].value_counts()[0:88].
 ↪to_frame(name='frequency')
Berlin_places2eat = Berlin_places2eat.reset_index()
Berlin_places2eat.rename(index=str, columns={"index": "Venue_Category",␣
 ↪"frequency": "Frequency"}, inplace=True)
Berlin_places2eat
```

```
[44]:         Venue_Category  Frequency
      0                Bakery         43
      1      Doner Restaurant         30
      2                  Café         29
      3           Coffee Shop         26
      4    Italian Restaurant         24
      ..                  …          …
      80            Wine Shop          1
      81            Juice Bar          1
      82    Frozen Yogurt Shop          1
      83     Food & Drink Shop          1
      84           Snack Place          1

      [85 rows x 2 columns]
```

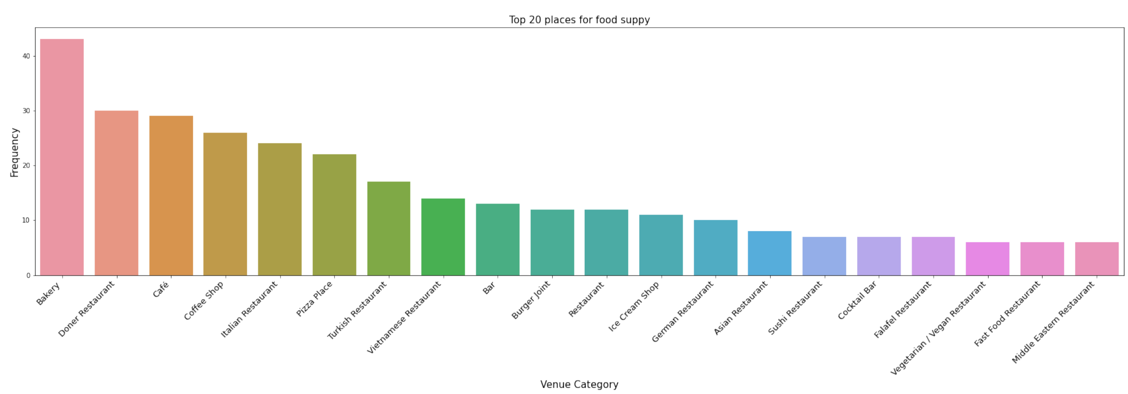
1.5.1 Various kinds of food places top the list of most common venues in Berlin tram stations. Organic and vegetarian food and beverages are not to be found within the top categories.

```
[45]: fig1 = plt.figure(figsize=(30,7))
s=sns.barplot(x="Venue_Category", y="Frequency", data=Berlin_places2eat)
s.set_xticklabels(s.get_xticklabels(), rotation=45,␣
 ↪horizontalalignment='right', fontsize=13)

plt.title('Most frequently places for food supply around stations', fontsize=15)
plt.xlabel("Venue Category", fontsize=15)
plt.ylabel ("Frequency", fontsize=15)
plt.savefig("Most_Freq_places2eat_top100.png", dpi=300)

plt.show()
```



[46]:
```python
# create a dataframe of top 20 categories
Berlin_places2eat_top20 = station_venues_toeat['Venue Category'].value_counts()[0:20].to_frame(name='frequency')
Berlin_places2eat_top20 = Berlin_places2eat_top20.reset_index()
Berlin_places2eat_top20.rename(index=str, columns={"index": "Venue_Category", "frequency": "Frequency"}, inplace=True)
```

[47]:
```python
fig2 = plt.figure(figsize=(30,7))
s=sns.barplot(x="Venue_Category", y="Frequency", data=Berlin_places2eat_top20)
s.set_xticklabels(s.get_xticklabels(), rotation=45, horizontalalignment='right', fontsize=13)

plt.title('Top 20 places for food suppy', fontsize=15)
plt.xlabel("Venue Category", fontsize=15)
plt.ylabel ("Frequency", fontsize=15)
plt.savefig("Most_Freq_places2eat_top20.png", dpi=300)

plt.show()
```



1.6 Determine Top 5 venues for each station

[48]:
```
print('There are {} uniques categories for all stations of Berlin.'.
  ↪format(len(station_venues_toeat['Venue Category'].unique())))
print('\nVenues with their total amount returned for each station: ')
station_venues_toeat.groupby('Station')['Venue'].count()
```

```
There are 85 uniques categories for all stations of Berlin.

Venues with their total amount returned for each station:
```

[48]:
```
Station
Adenauerplatz          7
Afrikanische Straße    3
Alexanderplatz         3
Alt-Mariendorf         2
Alt-Tegel              1
                      ..
Wittenau               2
Wittenbergplatz        3
Yorckstraße            2
Zitadelle              2
Zoologischer Garten    1
Name: Venue, Length: 133, dtype: int64
```

1.6.1 One Hot encoding

One-hot encoding will help to convert categorical variables (i.e., venues) into numeric variables. In this case, I will take the mean of the frequency of venue occurrence within a station.

[49]:
```
# one hot encoding of venue categories columns:https://pandas.pydata.org/
  ↪pandas-docs/stable/reference/api/pandas.get_dummies.html
station_onehot = pd.get_dummies(station_venues_toeat[['Venue Category']],␣
  ↪prefix= "", prefix_sep= " ")
# add neighborhood column back to dataframe
station_onehot['Station'] = station_venues_toeat['Station']
# move neighborhood column to the first column
fixed_columns = [station_onehot.columns[-1]] + list(station_onehot.columns[:-1])
station_onehot = station_onehot[fixed_columns]
print(station_onehot.shape)
station_onehot.head(100)
```

```
(444, 86)
```

[49]:
```
                Station   African Restaurant   Argentinian Restaurant  \
0         Adenauerplatz                    0                        0
1         Adenauerplatz                    0                        0
3         Adenauerplatz                    0                        0
5         Adenauerplatz                    0                        0
```

```
6          Adenauerplatz                    0                         0
..                     …                 …                       …
166  Franz-Neumann-Platz                    0                         0
167  Franz-Neumann-Platz                    0                         0
168  Franz-Neumann-Platz                    0                         0
169  Französische Straße                    0                         0
172  Französische Straße                    0                         0

      Asian Restaurant   BBQ Joint   Bakery   Bar   Beer Bar   Beer Store  \
0                    0           0        0     0          0            0
1                    0           0        0     0          0            0
3                    0           0        0     0          0            0
5                    0           0        0     0          0            0
6                    0           0        0     0          0            0
..                 …           …       …    …         …            …
166                  0           0        0     0          0            0
167                  0           0        0     0          0            0
168                  0           0        0     1          0            0
169                  0           0        0     0          0            0
172                  0           0        0     0          0            0

      Bistro  …  Tapas Restaurant   Taverna   Thai Restaurant  \
0          0  …                 0         0                 0
1          0  …                 0         0                 0
3          0  …                 0         0                 0
5          1  …                 0         0                 0
6          0  …                 0         0                 0
..       …  …               …       …               …
166        0  …                 0         0                 0
167        0  …                 0         0                 0
168        0  …                 0         0                 0
169        0  …                 0         0                 0
172        0  …                 0         0                 0

      Trattoria/Osteria   Turkish Restaurant   Ukrainian Restaurant  \
0                     0                    0                      0
1                     0                    0                      0
3                     0                    0                      0
5                     0                    0                      0
6                     0                    0                      0
..                  …                  …                    …
166                   0                    0                      0
167                   0                    0                      0
168                   0                    0                      0
169                   0                    0                      0
172                   0                    0                      0
```

```
     Vegetarian / Vegan Restaurant  Vietnamese Restaurant  Wine Bar  \
0                                0                      0         0
1                                0                      0         0
3                                0                      0         0
5                                0                      0         0
6                                0                      0         0
..                             …                      …       …
166                              0                      0         0
167                              0                      0         0
168                              0                      0         0
169                              0                      0         0
172                              0                      0         0

     Wine Shop
0            0
1            0
3            0
5            0
6            0
..         …
166          0
167          0
168          0
169          0
172          0

[100 rows x 86 columns]
```

[50]:
```
station_grouped = station_onehot.groupby('Station').mean().reset_index()
print(station_grouped.shape)
station_grouped.head()
```

```
(133, 86)
```

[50]:
```
               Station  African Restaurant  Argentinian Restaurant  \
0        Adenauerplatz                 0.0                0.000000
1  Afrikanische Straße                 0.0                0.333333
2       Alexanderplatz                 0.0                0.000000
3       Alt-Mariendorf                 0.0                0.000000
4            Alt-Tegel                 0.0                0.000000

   Asian Restaurant  BBQ Joint    Bakery  Bar  Beer Bar  Beer Store  \
0               0.0        0.0  0.142857  0.0       0.0         0.0
1               0.0        0.0  0.333333  0.0       0.0         0.0
2               0.0        0.0  0.000000  0.0       0.0         0.0
3               0.0        0.0  0.000000  0.0       0.0         0.0
4               0.0        0.0  0.000000  0.0       0.0         0.0
```

```
     Bistro  ...  Tapas Restaurant  Taverna  Thai Restaurant  \
0  0.142857  ...               0.0      0.0              0.0
1  0.000000  ...               0.0      0.0              0.0
2  0.000000  ...               0.0      0.0              0.0
3  0.000000  ...               0.0      0.0              0.0
4  0.000000  ...               0.0      0.0              0.0

   Trattoria/Osteria  Turkish Restaurant  Ukrainian Restaurant  \
0                0.0                 0.0                   0.0
1                0.0                 0.0                   0.0
2                0.0                 0.0                   0.0
3                0.0                 0.0                   0.0
4                0.0                 0.0                   0.0

   Vegetarian / Vegan Restaurant  Vietnamese Restaurant  Wine Bar  \
0                            0.0                    0.0       0.0
1                            0.0                    0.0       0.0
2                            0.0                    0.0       0.0
3                            0.0                    0.0       0.0
4                            0.0                    0.0       0.0

   Wine Shop
0        0.0
1        0.0
2        0.0
3        0.0
4        0.0

[5 rows x 86 columns]
```

The following cells contains a function that will help to sort venues of each station. In this analysis, the 5 most common venues each are taken under consideration.

[51]:
```
def return_most_common_venues(row, num_top_venues):
    row = row.iloc[1:]
    row_sorted = row.sort_values(ascending=False)
    return row_sorted.index.values[0:num_top_venues]
```

[52]:
```
num_top_venues = 5
indicators = ['st', 'nd', 'rd']
# create columns according to number of top venues
columns = ['Station']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))
# create a new dataframe
```

```
venues_sorted_station = pd.DataFrame(columns=columns)
# Put in here the above generated data
venues_sorted_station['Station'] = station_grouped['Station']
for ind in np.arange(station_grouped.shape[0]):
    venues_sorted_station.iloc[ind, 1:] =␣
 ↪return_most_common_venues(station_grouped.iloc[ind, :], num_top_venues)
```

[53]:
```
venues_sorted_station.head(15)
```

[53]:
```
                 Station    1st Most Common Venue  \
0         Adenauerplatz                Steakhouse
1     Afrikanische Straße               Hookah Bar
2        Alexanderplatz               Coffee Shop
3        Alt-Mariendorf         Greek Restaurant
4             Alt-Tegel         Doner Restaurant
5         Alt-Tempelhof     Vietnamese Restaurant
6       Altstadt Spandau              Cocktail Bar
7         Amrumer Straße               Coffee Shop
8      Augsburger Straße        Turkish Restaurant
9      Bayerischer Platz                 Wine Shop
10       Berliner Straße   Argentinian Restaurant
11       Bernauer Straße               Coffee Shop
12          Birkenstraße                      Café
13        Bismarckstraße         Asian Restaurant
14          Blissestraße     Vietnamese Restaurant

             2nd Most Common Venue 3rd Most Common Venue  \
0                      Coffee Shop          Cocktail Bar
1           Argentinian Restaurant                Bakery
2                 Doner Restaurant           Burger Joint
3               Italian Restaurant              Wine Shop
4                        Wine Shop                   Food
5                 Doner Restaurant    Fried Chicken Joint
6                           Bakery              Wine Shop
7            Vietnamese Restaurant                   Food
8                             Café     Spanish Restaurant
9                           Bakery      German Restaurant
10              Chinese Restaurant              Wine Shop
11    Vegetarian / Vegan Restaurant      Doner Restaurant
12    Vegetarian / Vegan Restaurant     German Restaurant
13                            Café              Wine Shop
14                Doner Restaurant                 Bakery

   4th Most Common Venue 5th Most Common Venue
0                 Bakery    Italian Restaurant
1        Food & Drink Shop             Dive Bar
2              Wine Shop      Currywurst Joint
```

```
3                   Food              Dive Bar
4           Dessert Shop              Dive Bar
5              Wine Shop                  Food
6                   Food              Dive Bar
7           Dessert Shop              Dive Bar
8              Wine Shop     Fish & Chips Shop
9        Currywurst Joint          Coffee Shop
10                  Food              Dive Bar
11                   Bar     Food & Drink Shop
12      Doner Restaurant            Restaurant
13                  Food              Dive Bar
14          Burger Joint     Sushi Restaurant
```

[54]:
```
#venues_sorted_station.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 133 entries, 0 to 132
Data columns (total 6 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Station                 133 non-null    object
 1   1st Most Common Venue  133 non-null    object
 2   2nd Most Common Venue  133 non-null    object
 3   3rd Most Common Venue  133 non-null    object
 4   4th Most Common Venue  133 non-null    object
 5   5th Most Common Venue  133 non-null    object
dtypes: object(6)
memory usage: 6.4+ KB
```

[55]:
```
#station_grouped.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 133 entries, 0 to 132
Data columns (total 86 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Station                          133 non-null    object
 1    African Restaurant              133 non-null    float64
 2    Argentinian Restaurant          133 non-null    float64
 3    Asian Restaurant                133 non-null    float64
 4    BBQ Joint                       133 non-null    float64
 5    Bakery                          133 non-null    float64
 6    Bar                             133 non-null    float64
 7    Beer Bar                        133 non-null    float64
 8    Beer Store                      133 non-null    float64
 9    Bistro                          133 non-null    float64
 10   Bosnian Restaurant              133 non-null    float64
 11   Brasserie                       133 non-null    float64
```

```
12   Brazilian Restaurant           133 non-null    float64
13   Breakfast Spot                 133 non-null    float64
14   Burger Joint                   133 non-null    float64
15   Burrito Place                  133 non-null    float64
16   Café                           133 non-null    float64
17   Chinese Restaurant             133 non-null    float64
18   Cigkofte Place                 133 non-null    float64
19   Cocktail Bar                   133 non-null    float64
20   Coffee Shop                    133 non-null    float64
21   Creperie                       133 non-null    float64
22   Currywurst Joint               133 non-null    float64
23   Dessert Shop                   133 non-null    float64
24   Dive Bar                       133 non-null    float64
25   Doner Restaurant               133 non-null    float64
26   Donut Shop                     133 non-null    float64
27   Eastern European Restaurant    133 non-null    float64
28   Falafel Restaurant             133 non-null    float64
29   Fast Food Restaurant           133 non-null    float64
30   Fish & Chips Shop              133 non-null    float64
31   Food                           133 non-null    float64
32   Food & Drink Shop              133 non-null    float64
33   French Restaurant              133 non-null    float64
34   Fried Chicken Joint            133 non-null    float64
35   Frozen Yogurt Shop             133 non-null    float64
36   Gastropub                      133 non-null    float64
37   German Restaurant              133 non-null    float64
38   Gourmet Shop                   133 non-null    float64
39   Greek Restaurant               133 non-null    float64
40   Halal Restaurant               133 non-null    float64
41   Hookah Bar                     133 non-null    float64
42   Hotel Bar                      133 non-null    float64
43   Ice Cream Shop                 133 non-null    float64
44   Indian Restaurant              133 non-null    float64
45   Indonesian Restaurant          133 non-null    float64
46   Israeli Restaurant             133 non-null    float64
47   Italian Restaurant             133 non-null    float64
48   Japanese Restaurant            133 non-null    float64
49   Juice Bar                      133 non-null    float64
50   Kebab Restaurant               133 non-null    float64
51   Korean Restaurant              133 non-null    float64
52   Kumpir Restaurant              133 non-null    float64
53   Kurdish Restaurant             133 non-null    float64
54   Lebanese Restaurant            133 non-null    float64
55   Mediterranean Restaurant       133 non-null    float64
56   Mexican Restaurant             133 non-null    float64
57   Middle Eastern Restaurant      133 non-null    float64
58   Modern European Restaurant     133 non-null    float64
59   Persian Restaurant             133 non-null    float64
```

```
 60   Pizza Place                     133 non-null    float64
 61   Pub                             133 non-null    float64
 62   Restaurant                      133 non-null    float64
 63   Russian Restaurant              133 non-null    float64
 64   Sandwich Place                  133 non-null    float64
 65   Schnitzel Restaurant            133 non-null    float64
 66   Seafood Restaurant              133 non-null    float64
 67   Shopping Mall                   133 non-null    float64
 68   Silesian Restaurant             133 non-null    float64
 69   Snack Place                     133 non-null    float64
 70   Spanish Restaurant              133 non-null    float64
 71   Sports Bar                      133 non-null    float64
 72   Steakhouse                      133 non-null    float64
 73   Sushi Restaurant                133 non-null    float64
 74   Taco Place                      133 non-null    float64
 75   Taiwanese Restaurant            133 non-null    float64
 76   Tapas Restaurant                133 non-null    float64
 77   Taverna                         133 non-null    float64
 78   Thai Restaurant                 133 non-null    float64
 79   Trattoria/Osteria               133 non-null    float64
 80   Turkish Restaurant              133 non-null    float64
 81   Ukrainian Restaurant            133 non-null    float64
 82   Vegetarian / Vegan Restaurant   133 non-null    float64
 83   Vietnamese Restaurant           133 non-null    float64
 84   Wine Bar                        133 non-null    float64
 85   Wine Shop                       133 non-null    float64
dtypes: float64(85), object(1)
memory usage: 89.5+ KB
```

1.7 Cluster of similar developed stations on venues similarity

The stations will be clustered or segmented based on a set of similar characteristics or features, i.e., their surrounding venues. K-Means clustering, which is used in this part of the analysis, is a machine learning algorithm that creates homogeneous subgroups/clusters from unlabeled data such that data points in each cluster are as similar as possible to each other according to a similarity measure (e.g., Euclidian distance).

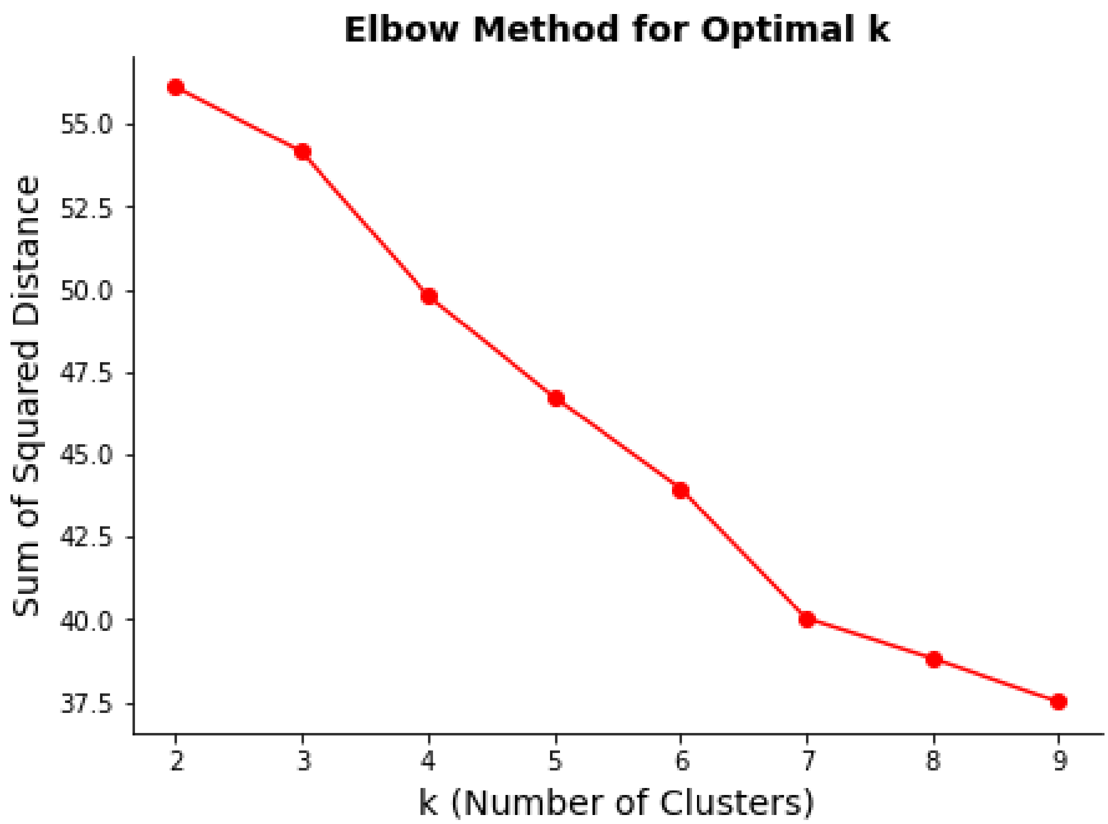
1.7.1 K-Means Clustering

Selecting the features (X): all venue category columns from the one-hot encoding dataframe.

[56]:
```
X = station_grouped.drop('Station', axis = 1)  # Select features
```

Determination of k (Ellbow methode) Before proceeding, a value of k (number of clusters) needs to be determined. The Elbow Method below calculates the sum of squared distances of data points to their closest centroid (cluster center) for different values of k. The optimal value of k is the one after which there is a plateau (no significant decrease in sum of squared distances).

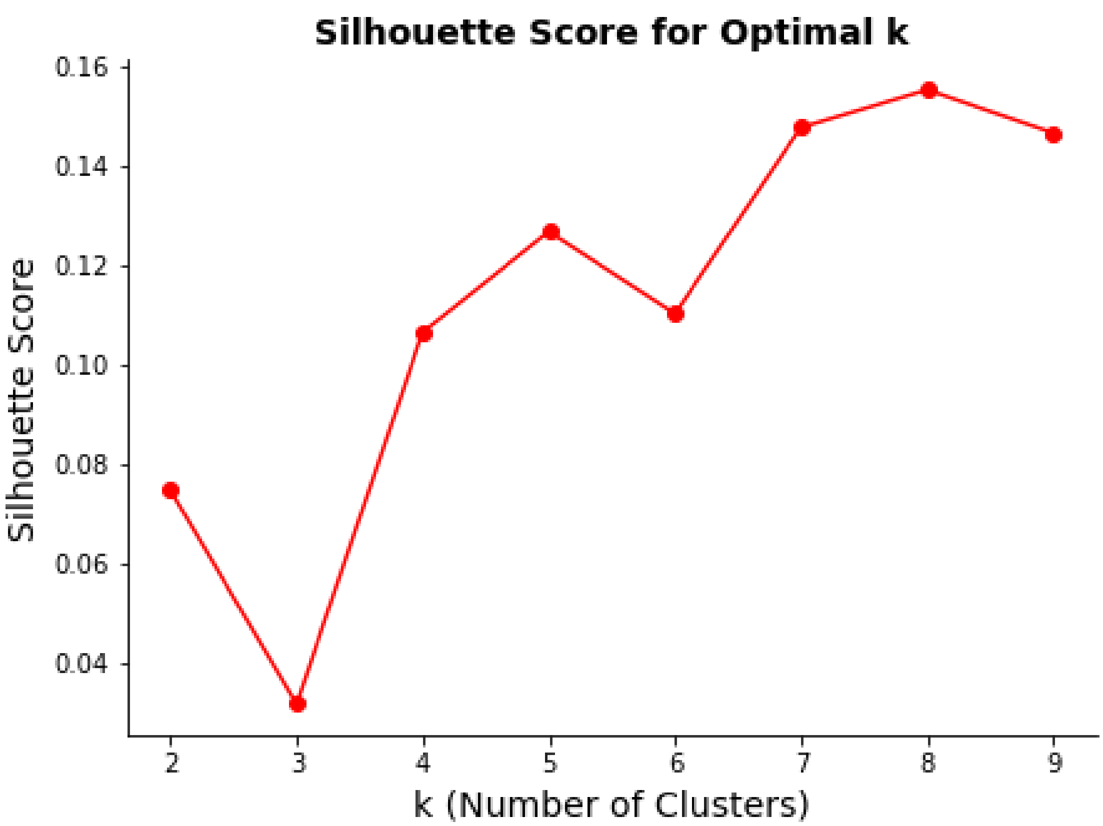
[57]:
```python
k_range = range(2,10)  # Range of k values to test
ssd = []  # Sum of Squared Distance

for k in k_range:
    model = KMeans(n_clusters=k, random_state=0).fit(X)
    ssd.append(model.inertia_)

plt.figure(figsize=(7,5))
plt.plot(k_range, ssd, 'ro-')
plt.title('Elbow Method for Optimal k', size=14, weight='bold')
plt.xlabel('k (Number of Clusters)', size=14)
plt.ylabel('Sum of Squared Distance', size=14)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.savefig('elbow.png', dpi=300, bbox_inches='tight')
plt.show()
```



Because there is no discernible "elbow" from the plot, another measure was applied: Silhouette Score.

[58]:

```
k_silh = range(2,10)
silh = []

for k in k_silh:
    model = KMeans(n_clusters=k, random_state=0).fit(X)
    labels = model.labels_
    silh.append(silhouette_score(X, labels, metric='euclidean'))

plt.figure(figsize=(7,5))
plt.plot(k_silh, silh, 'ro-')
plt.title('Silhouette Score for Optimal k', size=14, weight='bold')
plt.xlabel('k (Number of Clusters)', size=14)
plt.ylabel('Silhouette Score', size=14)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.savefig('silhouette.png', dpi=300, bbox_inches='tight')
plt.show()
```



Silhouette score varies from -1 to 1. A score value of 1 means the cluster is dense and well-separated from other clusters. A value nearing 0 represents overlapping clusters, data points are close to the decision boundary of neighboring clusters. A negative score indicates that the samples might have been assigned into the wrong clusters.

From the plot above, there is a peak at k=5 with which I'll proceed with that value as the number of optimal clusters. However, both methodes the ellbow and silhoutte, are not very clearly and need further investigation.

[59]:
```
# Kmean clustering to cluster the neigborhood
k = 5
kmeans = KMeans(n_clusters = k, random_state=0)
kmeans.fit(X)
```

[59]:
```
KMeans(n_clusters=5, random_state=0)
```

[60]:
```
# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

[60]:
```
array([3, 3, 4, 2, 4, 4, 0, 3, 3, 3])
```

[61]:
```
# add clustering labels
venues_sorted_station['Cluster_Labels'] = kmeans.labels_
```

[62]:
```
venues_sorted_station
```

[62]:
```
                  Station 1st Most Common Venue    2nd Most Common Venue  \
0           Adenauerplatz             Steakhouse              Coffee Shop
1      Afrikanische Straße             Hookah Bar  Argentinian Restaurant
2          Alexanderplatz            Coffee Shop        Doner Restaurant
3          Alt-Mariendorf       Greek Restaurant      Italian Restaurant
4               Alt-Tegel       Doner Restaurant               Wine Shop
..                    …                      …                       …
128              Wittenau       Doner Restaurant                  Bakery
129        Wittenbergplatz           Gourmet Shop      Turkish Restaurant
130            Yorckstraße               Dive Bar                  Bakery
131             Zitadelle          Shopping Mall   Fast Food Restaurant
132  Zoologischer Garten     Fried Chicken Joint               Wine Shop

    3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
0             Cocktail Bar                Bakery    Italian Restaurant
1                   Bakery      Food & Drink Shop              Dive Bar
2             Burger Joint             Wine Shop       Currywurst Joint
3                Wine Shop                  Food               Dive Bar
4                     Food          Dessert Shop               Dive Bar
..                     …                     …                     …
128              Wine Shop     Food & Drink Shop               Dive Bar
129          Burrito Place             Wine Shop                   Food
130              Wine Shop     Food & Drink Shop      Doner Restaurant
131              Wine Shop              Creperie          Dessert Shop
132                   Food          Dessert Shop               Dive Bar
```

```
     Cluster_Labels
0                 3
1                 3
2                 4
3                 2
4                 4
..              …
128               0
129               3
130               0
131               3
132               3

[133 rows x 7 columns]
```

[63]:

```python
data_station_cluster_merged = data_stat
# merge top venues_sorted with Berlin data_stat from the beginning
data_station_cluster_merged = venues_sorted_station.
 ↪join(data_station_cluster_merged.set_index('Station'), on='Station')
data_station_cluster_merged.head(50) # check the last columns
```

[63]:

```
                      Station       1st Most Common Venue  \
0               Adenauerplatz                   Steakhouse
1         Afrikanische Straße                   Hookah Bar
2              Alexanderplatz                  Coffee Shop
3              Alt-Mariendorf             Greek Restaurant
4                   Alt-Tegel             Doner Restaurant
5               Alt-Tempelhof        Vietnamese Restaurant
6             Altstadt Spandau                 Cocktail Bar
7               Amrumer Straße                  Coffee Shop
8            Augsburger Straße           Turkish Restaurant
9            Bayerischer Platz                    Wine Shop
10             Berliner Straße      Argentinian Restaurant
11             Bernauer Straße                  Coffee Shop
12               Birkenstraße                         Café
13             Bismarckstraße             Asian Restaurant
14               Blissestraße        Vietnamese Restaurant
15               Boddinstraße                  Pizza Place
16                Borsigwerke           Italian Restaurant
17           Brandenburger Tor                    Hotel Bar
18           Breitenbachplatz                 Dessert Shop
19                  Britz-Süd                       Bakery
20                Bundesplatz    Middle Eastern Restaurant
21                Dahlem-Dorf                  Pizza Place
22              Deutsche Oper           Italian Restaurant
23          Eberswalder Straße                       Bakery
```

24	Eisenacher Straße	Vietnamese Restaurant
25	Elsterwerdaer Platz	Asian Restaurant
26	Frankfurter Allee	Bakery
27	Franz-Neumann-Platz	Hookah Bar
28	Französische Straße	Gourmet Shop
29	Friedrich-Wilhelm-Platz	Spanish Restaurant
30	Friedrichstraße	German Restaurant
31	Gneisenaustraße	Kebab Restaurant
32	Grenzallee	Bakery
33	Görlitzer Bahnhof	Turkish Restaurant
34	Güntzelstraße	Indian Restaurant
35	Halemweg	Pizza Place
36	Hallesches Tor	BBQ Joint
37	Hansaplatz	Turkish Restaurant
38	Hauptbahnhof	Coffee Shop
39	Hausvogteiplatz	Coffee Shop
40	Heinrich-Heine-Straße	Doner Restaurant
41	Hermannplatz	Sandwich Place
42	Hermannstraße	Doner Restaurant
43	Hohenzollernplatz	Restaurant
44	Holzhauser Straße	Doner Restaurant
45	Innsbrucker Platz	Seafood Restaurant
46	Jannowitzbrücke	Turkish Restaurant
47	Johannisthaler Chaussee	Ice Cream Shop
48	Jungfernheide	Sandwich Place
49	Kaiserdamm	Vietnamese Restaurant

	2nd Most Common Venue	3rd Most Common Venue \
0	Coffee Shop	Cocktail Bar
1	Argentinian Restaurant	Bakery
2	Doner Restaurant	Burger Joint
3	Italian Restaurant	Wine Shop
4	Wine Shop	Food
5	Doner Restaurant	Fried Chicken Joint
6	Bakery	Wine Shop
7	Vietnamese Restaurant	Food
8	Café	Spanish Restaurant
9	Bakery	German Restaurant
10	Chinese Restaurant	Wine Shop
11	Vegetarian / Vegan Restaurant	Doner Restaurant
12	Vegetarian / Vegan Restaurant	German Restaurant
13	Café	Wine Shop
14	Doner Restaurant	Bakery
15	Dive Bar	Bar
16	Wine Shop	Food & Drink Shop
17	Restaurant	Coffee Shop
18	Mexican Restaurant	Food

```
19                Wine Shop              Food & Drink Shop
20       Mexican Restaurant                      Wine Shop
21                   Bakery                   Burger Joint
22       Chinese Restaurant                      Wine Shop
23    Vietnamese Restaurant              Indian Restaurant
24                   Bakery                      Wine Shop
25         Doner Restaurant             Italian Restaurant
26              Coffee Shop             Italian Restaurant
27              Pizza Place                            Bar
28               Restaurant             Italian Restaurant
29       Mexican Restaurant                      Wine Shop
30    Vietnamese Restaurant   Vegetarian / Vegan Restaurant
31                Wine Shop              Food & Drink Shop
32                Wine Shop              Food & Drink Shop
33       African Restaurant                            Bar
34                      Bar                      Wine Shop
35       Persian Restaurant                         Bakery
36                   Bakery                         Bistro
37         Kebab Restaurant                      Wine Shop
38                   Bakery                 Sandwich Place
39          Thai Restaurant                     Restaurant
40                Wine Shop                           Food
41              Coffee Shop                   Gourmet Shop
42           Ice Cream Shop                 Breakfast Spot
43                     Food                   Dessert Shop
44                Wine Shop                           Food
45     Ukrainian Restaurant                      Wine Shop
46                Wine Shop                           Food
47                     Food                   Dessert Shop
48                   Bakery            Fast Food Restaurant
49               Donut Shop                      Wine Shop

          4th Most Common Venue  5th Most Common Venue  Cluster_Labels  \
0                        Bakery     Italian Restaurant               3
1              Food & Drink Shop              Dive Bar               3
2                     Wine Shop       Currywurst Joint               4
3                          Food               Dive Bar               2
4                  Dessert Shop               Dive Bar               4
5                     Wine Shop                   Food               4
6                          Food               Dive Bar               0
7                  Dessert Shop               Dive Bar               3
8                     Wine Shop      Fish & Chips Shop               3
9              Currywurst Joint            Coffee Shop               3
10                         Food               Dive Bar               3
11                          Bar      Food & Drink Shop               3
12             Doner Restaurant             Restaurant               3
13                         Food               Dive Bar               3
```

14	Burger Joint	Sushi Restaurant	3
15	Thai Restaurant	Shopping Mall	3
16	Dive Bar	Doner Restaurant	2
17	Modern European Restaurant	Donut Shop	3
18	Dive Bar	Doner Restaurant	1
19	Dive Bar	Doner Restaurant	0
20	Fish & Chips Shop	Dive Bar	1
21	Wine Shop	Food	3
22	Food	Dive Bar	2
23	Cocktail Bar	Taco Place	3
24	Food & Drink Shop	Dive Bar	0
25	Wine Shop	Food & Drink Shop	4
26	Kebab Restaurant	Fast Food Restaurant	3
27	Café	Food	3
28	Fish & Chips Shop	Dessert Shop	2
29	Creperie	Dessert Shop	1
30	Bakery	Wine Shop	3
31	Dive Bar	Doner Restaurant	3
32	Dive Bar	Doner Restaurant	0
33	Coffee Shop	Doner Restaurant	3
34	Food & Drink Shop	Dive Bar	3
35	Wine Shop	Fish & Chips Shop	3
36	Wine Shop	Food & Drink Shop	3
37	Food	Dessert Shop	3
38	Ice Cream Shop	Sushi Restaurant	3
39	Gourmet Shop	Fast Food Restaurant	3
40	Dessert Shop	Dive Bar	4
41	Creperie	Fish & Chips Shop	3
42	Food	Dive Bar	4
43	Dive Bar	Doner Restaurant	2
44	Dessert Shop	Dive Bar	4
45	Fish & Chips Shop	Dessert Shop	3
46	Dessert Shop	Dive Bar	3
47	Dive Bar	Doner Restaurant	3
48	Food	Dessert Shop	3
49	Food	Dessert Shop	3

	Locality	Latitude	Longitude
0	Charlottenburg	52.499722	13.307222
1	Wedding	52.560556	13.334167
2	Mitte	52.521389	13.413333
3	Mariendorf	52.439722	13.387500
4	Tegel	52.589444	13.283611
5	Tempelhof	52.466111	13.385556
6	Spandau	52.539167	13.205556
7	Wedding	52.542222	13.348889
8	Charlottenburg	52.500556	13.336389

9	Schöneberg	52.488611	13.340000
10	Wilmersdorf	52.487222	13.330833
11	Mitte	52.537500	13.396667
12	Moabit	52.532222	13.341389
13	Charlottenburg	52.511389	13.304722
14	Wilmersdorf	52.486667	13.321944
15	Neukölln	52.479444	13.425556
16	Tegel	52.581944	13.290833
17	Mitte	52.516389	13.380833
18	Dahlem	52.466944	13.308611
19	Britz	52.437778	13.448333
20	Wilmersdorf	52.478889	13.328056
21	Dahlem	52.457500	13.289722
22	Charlottenburg	52.511944	13.310556
23	Prenzlauer Berg	52.541667	13.412222
24	Schöneberg	52.489444	13.350278
25	Biesdorf	52.505000	13.560556
26	Friedrichshain	52.515000	13.474722
27	Reinickendorf	52.563889	13.364167
28	Mitte	52.514722	13.389167
29	Friedenau	52.471944	13.328611
30	Mitte	52.520278	13.386944
31	Kreuzberg	52.491389	13.396111
32	Britz	52.463333	13.443889
33	Kreuzberg	52.499167	13.428056
34	Wilmersdorf	52.491944	13.330833
35	Charlottenburg-Nord	52.536667	13.286389
36	Kreuzberg	52.497778	13.391111
37	Hansaviertel	52.517778	13.342222
38	Moabit	52.525000	13.369444
39	Mitte	52.513056	13.396667
40	Mitte	52.510278	13.415833
41	Neukölln	52.487222	13.424722
42	Neukölln	52.467500	13.431389
43	Wilmersdorf	52.494167	13.324722
44	Tegel	52.575833	13.296111
45	Schöneberg	52.478611	13.343889
46	Mitte	52.515000	13.418056
47	Gropiusstadt	52.429444	13.453056
48	Charlottenburg	52.530833	13.300833
49	Westend	52.510000	13.282222

1.8 Visualizing Clusters

Now that each station has been assigned a cluster label, it would be helpful to visualize the clusters on a map of Berlin to see how they are distributed. Folium library is used for this purpose.

[64]:
```
# create map
map_clustered = folium.Map(location=[latitude, longitude], zoom_start=12)
# set color scheme for the clusters
x = np.arange(k)
ys = [i + x + (i*x)**2 for i in range(k)]
#colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
colors_array = cm.jet(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]
# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(data_station_cluster_merged['Latitude'],␣
 ↪data_station_cluster_merged['Longitude'],
                                   data_station_cluster_merged['Station'],␣
 ↪data_station_cluster_merged['Cluster_Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker([lat,␣
 ↪lon],radius=5,popup=label,color=rainbow[cluster-1],fill=True,fill_color=rainbow[cluster-1],f
 ↪9).add_to(map_clustered)
map_clustered
```

[64]: <folium.folium.Map at 0xc783a00>

2 Examining Each Cluster

Each cluster is filtered from the dataframe previously created in the clustering stage. The clusters are separately analyzed in order to gain an understanding of a discriminating venue that characterize each of them. Means, the 1st and 2nd most common venue category from each cluster will be singled out.

2.1 Cluster 0

Color code in map: wine red (or brown for some eyes)

[65]:
```
cluster0 = data_station_cluster_merged.
 ↪loc[data_station_cluster_merged['Cluster_Labels'] == 0,␣
 ↪data_station_cluster_merged.columns[[0] + list(range(1,␣
 ↪data_station_cluster_merged.shape[1]))]]
cluster0
```

[65]:
```
                      Station    1st Most Common Venue 2nd Most Common Venue  \
6            Altstadt Spandau              Cocktail Bar                Bakery
19                  Britz-Süd                    Bakery             Wine Shop
24          Eisenacher Straße    Vietnamese Restaurant                Bakery
32                 Grenzallee                    Bakery             Wine Shop
50    Kaiserin-Augusta-Straße                    Bakery           Coffee Shop
65           Magdalenenstraße                    Bakery             Wine Shop
74               Nauener Platz         Doner Restaurant                Bakery
```

```
84              Otisstraße              Bakery          Wine Shop
123            Voltastraße              Bakery                Bar
124      Warschauer Straße              Bakery          Wine Shop
126              Westhafen         Coffee Shop             Bakery
128               Wittenau    Doner Restaurant             Bakery
130            Yorckstraße            Dive Bar             Bakery

    3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
6               Wine Shop                  Food              Dive Bar
19      Food & Drink Shop              Dive Bar      Doner Restaurant
24              Wine Shop     Food & Drink Shop              Dive Bar
32      Food & Drink Shop              Dive Bar      Doner Restaurant
50              Wine Shop                  Food              Dive Bar
65      Food & Drink Shop              Dive Bar      Doner Restaurant
74              Wine Shop     Food & Drink Shop              Dive Bar
84      Food & Drink Shop              Dive Bar      Doner Restaurant
123             Gastropub             Wine Shop                  Food
124     Food & Drink Shop              Dive Bar      Doner Restaurant
126             Wine Shop                  Food              Dive Bar
128             Wine Shop     Food & Drink Shop              Dive Bar
130             Wine Shop     Food & Drink Shop      Doner Restaurant

     Cluster_Labels        Locality   Latitude  Longitude
6                 0         Spandau  52.539167  13.205556
19                0           Britz  52.437778  13.448333
24                0      Schöneberg  52.489444  13.350278
32                0           Britz  52.463333  13.443889
50                0       Tempelhof  52.460000  13.384722
65                0     Lichtenberg  52.512500  13.486389
74                0         Wedding  52.551667  13.367500
84                0   Reinickendorf  52.571111  13.302778
123               0   Gesundbrunnen  52.542222  13.393056
124               0  Friedrichshain  52.505278  13.449167
126               0          Moabit  52.536389  13.343889
128               0        Wittenau  52.595833  13.336667
130               0      Schöneberg  52.493056  13.370833
```

[66]:
```
cluster0.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13 entries, 6 to 130
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Station                13 non-null     object
 1   1st Most Common Venue  13 non-null     object
 2   2nd Most Common Venue  13 non-null     object
 3   3rd Most Common Venue  13 non-null     object
```

```
 4   4th Most Common Venue  13 non-null     object
 5   5th Most Common Venue  13 non-null     object
 6   Cluster_Labels         13 non-null     int32
 7   Locality               13 non-null     object
 8   Latitude               13 non-null     float64
 9   Longitude              13 non-null     float64
dtypes: float64(2), int32(1), object(7)
memory usage: 1.1+ KB
```

[67]:
```
# Filter the no. 1 most common venues in the cluster
top1_cluster0 = cluster0.iloc[:, 1].value_counts().reset_index()
top1_cluster0.columns = ['1st Most Common Venue', 'Count']
top1_cluster0
```

[67]:
```
   1st Most Common Venue  Count
0                 Bakery      7
1       Doner Restaurant      2
2  Vietnamese Restaurant      1
3           Cocktail Bar      1
4            Coffee Shop      1
5               Dive Bar      1
```

[68]:
```
# Filter the no. 2 most common venues in the cluster
top2_cluster0 = cluster0.iloc[:, 2].value_counts().reset_index()
top2_cluster0.columns = ['2st Most Common Venue', 'Count']
top2_cluster0
```

[68]:
```
  2st Most Common Venue  Count
0                Bakery      6
1             Wine Shop      5
2           Coffee Shop      1
3                   Bar      1
```

Observation for Cluster 0: Bakeries are the prominent venue in this cluster 0 containing 13 stations.

2.2 Cluster 1

Color code in map: dark blue

[69]:
```
cluster1 = data_station_cluster_merged.
 ↪loc[data_station_cluster_merged['Cluster_Labels'] == 1,␣
 ↪data_station_cluster_merged.columns[[0] + list(range(1,␣
 ↪data_station_cluster_merged.shape[1]))]]
cluster1
```

[69]:
```
                     Station       1st Most Common Venue  \
18          Breitenbachplatz                Dessert Shop
```

```
20             Bundesplatz    Middle Eastern Restaurant
29  Friedrich-Wilhelm-Platz           Spanish Restaurant

   2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue  \
18    Mexican Restaurant                  Food              Dive Bar
20    Mexican Restaurant             Wine Shop     Fish & Chips Shop
29    Mexican Restaurant             Wine Shop              Creperie

   5th Most Common Venue  Cluster_Labels     Locality   Latitude  Longitude
18      Doner Restaurant               1       Dahlem  52.466944  13.308611
20              Dive Bar               1  Wilmersdorf  52.478889  13.328056
29          Dessert Shop               1    Friedenau  52.471944  13.328611
```

[70]:

```
cluster1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3 entries, 18 to 29
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Station                3 non-null      object
 1   1st Most Common Venue  3 non-null      object
 2   2nd Most Common Venue  3 non-null      object
 3   3rd Most Common Venue  3 non-null      object
 4   4th Most Common Venue  3 non-null      object
 5   5th Most Common Venue  3 non-null      object
 6   Cluster_Labels         3 non-null      int32
 7   Locality               3 non-null      object
 8   Latitude               3 non-null      float64
 9   Longitude              3 non-null      float64
dtypes: float64(2), int32(1), object(7)
memory usage: 252.0+ bytes
```

Observation for Cluster 1: With 3 members (stations) the smallest cluster dominiated by restaurants (mexican) and wine shops.

2.3 Cluster 2

Color code in map: brighter blue

[71]:

```
cluster2 = data_station_cluster_merged.
↪loc[data_station_cluster_merged['Cluster_Labels'] == 2,␣
↪data_station_cluster_merged.columns[[0] + list(range(1,␣
↪data_station_cluster_merged.shape[1]))]]
cluster2
```

[71]:

```
                        Station 1st Most Common Venue 2nd Most Common Venue  \
3             Alt-Mariendorf       Greek Restaurant    Italian Restaurant
```

```
16          Borsigwerke     Italian Restaurant            Wine Shop
22         Deutsche Oper    Italian Restaurant   Chinese Restaurant
28     Französische Straße        Gourmet Shop           Restaurant
43       Hohenzollernplatz          Restaurant                 Food
61     Kurt-Schumacher-Platz        Restaurant     Doner Restaurant
89        Podbielskiallee           Restaurant                 Food
95      Richard-Wagner-Platz Italian Restaurant            Wine Shop
96               Rohrdamm   Italian Restaurant            Wine Shop
99          Rotes Rathaus   Italian Restaurant            Wine Shop
100               Rudow     Italian Restaurant            Wine Shop

   3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
3              Wine Shop                  Food              Dive Bar
16     Food & Drink Shop              Dive Bar      Doner Restaurant
22             Wine Shop                  Food              Dive Bar
28    Italian Restaurant     Fish & Chips Shop          Dessert Shop
43          Dessert Shop              Dive Bar      Doner Restaurant
61    Italian Restaurant                Bistro                  Food
89          Dessert Shop              Dive Bar      Doner Restaurant
95     Food & Drink Shop              Dive Bar      Doner Restaurant
96     Food & Drink Shop              Dive Bar      Doner Restaurant
99     Food & Drink Shop              Dive Bar      Doner Restaurant
100    Food & Drink Shop              Dive Bar      Doner Restaurant

     Cluster_Labels       Locality   Latitude  Longitude
3                 2     Mariendorf  52.439722  13.387500
16                2          Tegel  52.581944  13.290833
22                2  Charlottenburg 52.511944  13.310556
28                2          Mitte  52.514722  13.389167
43                2    Wilmersdorf  52.494167  13.324722
61                2   Reinickendorf 52.563333  13.327500
89                2         Dahlem  52.464167  13.295833
95                2  Charlottenburg 52.515833  13.307500
96                2    Siemensstadt 52.537222  13.262500
99                2          Mitte  52.518611  13.408333
100               2          Rudow  52.416111  13.495278
```

[72]:
```
cluster2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11 entries, 3 to 100
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Station                11 non-null     object
 1   1st Most Common Venue  11 non-null     object
 2   2nd Most Common Venue  11 non-null     object
 3   3rd Most Common Venue  11 non-null     object
```

```
 4   4th Most Common Venue  11 non-null     object
 5   5th Most Common Venue  11 non-null     object
 6   Cluster_Labels         11 non-null     int32
 7   Locality               11 non-null     object
 8   Latitude               11 non-null     float64
 9   Longitude              11 non-null     float64
dtypes: float64(2), int32(1), object(7)
memory usage: 924.0+ bytes
```

[73]:
```
# Filter the no. 1 most common venues in the cluster
top1_cluster2 = cluster2.iloc[:, 1].value_counts().reset_index()
top1_cluster2.columns = ['1st Most Common Venue', 'Count']
top1_cluster2
```

[73]:
```
  1st Most Common Venue  Count
0    Italian Restaurant      6
1            Restaurant      3
2           Gourmet Shop      1
3      Greek Restaurant      1
```

[74]:
```
# Filter the no. 2 most common venues in the cluster
top2_cluster2 = cluster2.iloc[:, 2].value_counts().reset_index()
top2_cluster2.columns = ['2st Most Common Venue', 'Count']
top2_cluster2
```

[74]:
```
  2st Most Common Venue  Count
0             Wine Shop      5
1                  Food      2
2      Doner Restaurant      1
3    Chinese Restaurant      1
4    Italian Restaurant      1
5            Restaurant      1
```

Observation for Cluster 2: Thirteen stations fall into this cluster. Dominated by italian restaurants and wine shops.

2.4 Cluster 3

Color code in map: bright green

[75]:
```
cluster3 = data_station_cluster_merged.
 ↪loc[data_station_cluster_merged['Cluster_Labels'] == 3,␣
 ↪data_station_cluster_merged.columns[[0] + list(range(1,␣
 ↪data_station_cluster_merged.shape[1]))]]
cluster3
```

[75]:
```
                     Station   1st Most Common Venue    2nd Most Common Venue  \
0            Adenauerplatz               Steakhouse               Coffee Shop
```

```
1     Afrikanische Straße               Hookah Bar   Argentinian Restaurant
7          Amrumer Straße              Coffee Shop   Vietnamese Restaurant
8        Augsburger Straße      Turkish Restaurant                    Café
9        Bayerischer Platz                Wine Shop                  Bakery
..                     …                        …                       …
125             Weberwiese    Vietnamese Restaurant                     Bar
127   Wilmersdorfer Straße              Pizza Place       Doner Restaurant
129         Wittenbergplatz             Gourmet Shop      Turkish Restaurant
131               Zitadelle            Shopping Mall    Fast Food Restaurant
132   Zoologischer Garten      Fried Chicken Joint               Wine Shop

    3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
0            Cocktail Bar                Bakery    Italian Restaurant
1                  Bakery     Food & Drink Shop              Dive Bar
7                    Food          Dessert Shop              Dive Bar
8      Spanish Restaurant             Wine Shop     Fish & Chips Shop
9       German Restaurant      Currywurst Joint           Coffee Shop
..                    …                     …                     …
125             Wine Shop     Food & Drink Shop              Dive Bar
127                Bakery    Italian Restaurant                  Café
129         Burrito Place             Wine Shop                  Food
131             Wine Shop              Creperie          Dessert Shop
132                  Food          Dessert Shop              Dive Bar

     Cluster_Labels        Locality   Latitude  Longitude
0                 3  Charlottenburg  52.499722  13.307222
1                 3         Wedding  52.560556  13.334167
7                 3         Wedding  52.542222  13.348889
8                 3  Charlottenburg  52.500556  13.336389
9                 3      Schöneberg  52.488611  13.340000
..              …             …         …          …
125               3  Friedrichshain  52.516667  13.445000
127               3  Charlottenburg  52.506667  13.306667
129               3      Schöneberg  52.501944  13.343056
131               3      Haselhorst  52.537778  13.217778
132               3  Charlottenburg  52.507222  13.332500

[92 rows x 10 columns]
```

[76]:

```python
# Filter the no. 1 most common venues in the cluster
top1_cluster3 = cluster3.iloc[:, 1].value_counts().reset_index()
top1_cluster3.columns = ['1st Most Common Venue', 'Count']
top1_cluster3
```

[76]:

```
  1st Most Common Venue  Count
0           Coffee Shop     10
1                  Café      9
```

```
2                Pizza Place      8
3              Ice Cream Shop      6
4          Turkish Restaurant      6
5                   Hotel Bar      5
6       Vietnamese Restaurant      4
7            Asian Restaurant      4
8           German Restaurant      3
9                         Pub      3
10                 Hookah Bar      2
11         Italian Restaurant      2
12             Sandwich Place      2
13                     Bakery      2
14                  Wine Shop      1
15                       Food      1
16           Kebab Restaurant      1
17          Food & Drink Shop      1
18                 Steakhouse      1
19              Shopping Mall      1
20     Argentinian Restaurant      1
21          Indian Restaurant      1
22                  BBQ Joint      1
23           Sushi Restaurant      1
24         Seafood Restaurant      1
25       Schnitzel Restaurant      1
26       Brazilian Restaurant      1
27               Burger Joint      1
28         Falafel Restaurant      1
29                 Sports Bar      1
30                 Brasserie      1
31             Cigkofte Place      1
32         Fried Chicken Joint      1
33         Persian Restaurant      1
34           Doner Restaurant      1
35               Gourmet Shop      1
36                 Restaurant      1
37         Chinese Restaurant      1
38               Cocktail Bar      1
39           Halal Restaurant      1
```

[77]:

```
# Filter the no. 2 most common venues in the cluster
top2_cluster3 = cluster3.iloc[:, 2].value_counts().reset_index()
top2_cluster3.columns = ['2st Most Common Venue', 'Count']
top2_cluster3
```

[77]:

	2st Most Common Venue	Count
0	Wine Shop	14
1	Vietnamese Restaurant	6

```
2                       Bakery      6
3                  Pizza Place      4
4            Trattoria/Osteria      4
5                  Coffee Shop      4
6                         Café      4
7             Doner Restaurant      3
8           Turkish Restaurant      3
9   Vegetarian / Vegan Restaurant   3
10                  Donut Shop      2
11                         Bar      2
12       Fast Food Restaurant       2
13           German Restaurant      2
14                    Dive Bar      2
15   Middle Eastern Restaurant      2
16                  Restaurant      2
17       Indonesian Restaurant      2
18                Cocktail Bar      2
19          Italian Restaurant      2
20              Ice Cream Shop      2
21      Argentinian Restaurant      1
22          Israeli Restaurant      1
23          African Restaurant      1
24  Eastern European Restaurant     1
25          Chinese Restaurant      1
26            Kebab Restaurant      1
27                    Wine Bar      1
28                  Sports Bar      1
29                         Pub      1
30        Ukrainian Restaurant      1
31            Sushi Restaurant      1
32                  Hookah Bar      1
33          Persian Restaurant      1
34                   Hotel Bar      1
35                     Taverna      1
36             Thai Restaurant      1
37              Sandwich Place      1
38           Korean Restaurant      1
39                        Food      1
```

[78]:
```
cluster3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 92 entries, 0 to 132
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Station                 92 non-null     object
 1   1st Most Common Venue   92 non-null     object
```

```
 2   2nd Most Common Venue  92 non-null     object
 3   3rd Most Common Venue  92 non-null     object
 4   4th Most Common Venue  92 non-null     object
 5   5th Most Common Venue  92 non-null     object
 6   Cluster_Labels         92 non-null     int32
 7   Locality               92 non-null     object
 8   Latitude               92 non-null     float64
 9   Longitude              92 non-null     float64
dtypes: float64(2), int32(1), object(7)
memory usage: 7.5+ KB
```

Observation for Cluster 3: Largest cluster with 92 member stations. Prominent are Coffee and Cafe places, followed by pizza and turkish food.

2.5 Cluster 4

Color code in map: orange

[79]:
```
cluster4 = data_station_cluster_merged.
 ↪loc[data_station_cluster_merged['Cluster_Labels'] == 4,␣
 ↪data_station_cluster_merged.columns[[0] + list(range(1,␣
 ↪data_station_cluster_merged.shape[1]))]]
cluster4
```

[79]:
```
                  Station   1st Most Common Venue 2nd Most Common Venue  \
2          Alexanderplatz             Coffee Shop      Doner Restaurant
4               Alt-Tegel        Doner Restaurant             Wine Shop
5           Alt-Tempelhof  Vietnamese Restaurant      Doner Restaurant
25     Elsterwerdaer Platz        Asian Restaurant      Doner Restaurant
40   Heinrich-Heine-Straße        Doner Restaurant             Wine Shop
42          Hermannstraße        Doner Restaurant        Ice Cream Shop
44       Holzhauser Straße        Doner Restaurant             Wine Shop
52          Kaulsdorf-Nord        Doner Restaurant    Turkish Restaurant
58            Krumme Lanke        Doner Restaurant    Italian Restaurant
60       Kurfürstenstraße        Asian Restaurant      Doner Restaurant
62            Leinestraße        Doner Restaurant                   Bar
66             Mehringdamm        Doner Restaurant             Wine Shop
81       Oskar-Helene-Heim        Doner Restaurant             Wine Shop
82           Osloer Straße        Doner Restaurant           Pizza Place

   3rd Most Common Venue  4th Most Common Venue 5th Most Common Venue  \
2            Burger Joint              Wine Shop      Currywurst Joint
4                    Food          Dessert Shop              Dive Bar
5     Fried Chicken Joint              Wine Shop                  Food
25     Italian Restaurant              Wine Shop     Food & Drink Shop
40                   Food          Dessert Shop              Dive Bar
42         Breakfast Spot                   Food              Dive Bar
44                   Food          Dessert Shop              Dive Bar
```

```
52              Wine Shop                   Food          Dessert Shop
58              Wine Shop      Food & Drink Shop              Dive Bar
60                 Bakery              Wine Shop     Food & Drink Shop
62    Bosnian Restaurant               Wine Shop     Food & Drink Shop
66                   Food           Dessert Shop              Dive Bar
81                   Food           Dessert Shop              Dive Bar
82                 Bakery   Fast Food Restaurant             Wine Shop

    Cluster_Labels       Locality   Latitude  Longitude
2                4          Mitte  52.521389  13.413333
4                4          Tegel  52.589444  13.283611
5                4      Tempelhof  52.466111  13.385556
25               4       Biesdorf  52.505000  13.560556
40               4          Mitte  52.510278  13.415833
42               4       Neukölln  52.467500  13.431389
44               4          Tegel  52.575833  13.296111
52               4    Hellersdorf  52.521111  13.588889
58               4     Zehlendorf  52.443333  13.241389
60               4     Tiergarten  52.500000  13.361944
62               4       Neukölln  52.473611  13.428056
66               4      Kreuzberg  52.494444  13.388611
81               4         Dahlem  52.450278  13.269722
82               4  Gesundbrunnen  52.556944  13.373333
```

[80]:
```
cluster4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14 entries, 2 to 82
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Station                 14 non-null     object
 1   1st Most Common Venue   14 non-null     object
 2   2nd Most Common Venue   14 non-null     object
 3   3rd Most Common Venue   14 non-null     object
 4   4th Most Common Venue   14 non-null     object
 5   5th Most Common Venue   14 non-null     object
 6   Cluster_Labels          14 non-null     int32
 7   Locality                14 non-null     object
 8   Latitude                14 non-null     float64
 9   Longitude               14 non-null     float64
dtypes: float64(2), int32(1), object(7)
memory usage: 1.1+ KB
```

[81]:
```
# Filter the no. 1 most common venues in the cluster
top1_cluster4 = cluster4.iloc[:, 1].value_counts().reset_index()
top1_cluster4.columns = ['1st Most Common Venue', 'Count']
top1_cluster4
```

```
[81]:      1st Most Common Venue  Count
      0          Doner Restaurant     10
      1          Asian Restaurant      2
      2     Vietnamese Restaurant      1
      3               Coffee Shop      1
```

```python
[82]: top2_cluster4 = cluster4.iloc[:, 2].value_counts().reset_index()
      top2_cluster4.columns = ['2st Most Common Venue', 'Count']
      top2_cluster4
```

```
[82]:    2st Most Common Venue  Count
      0              Wine Shop      5
      1       Doner Restaurant      4
      2     Turkish Restaurant      1
      3          Ice Cream Shop      1
      4     Italian Restaurant      1
      5                    Bar      1
      6            Pizza Place      1
```

Observation for Cluster 4: Fourteen entries fall into this cluster. Dominated by Doner restaurants and again wine shops.

3 Results and Discussion

Exploratory data analysis as well as machine learning and visualization techniques have provided us with some insights into the problem at hand. A total of 842 items originated by 184 venue catagories for all 178 Berlin metro stations regions were returned at the time the API call was made. The search radius was chosen quite narrow with 100 m. After removing venue cateories not of interest for the regarded food industry (such as the tram station itself, gym, IT-Sevices etc), 85 unique categories were being left after modification. The most common categories overall are 1. Bakeries, 2. Doner restaurants, 3. Cafe, 4. Coffee Shops, and 5. Italian and Pizza places.

After deciding on an optimal k value of 5, K-Means algorithm was run to cluster the stations based on their most common surrounding venues. To determine this optimal k value, two common methodes Elbow and silhouette were applied. The result of k = 5 result is ambiguous and needs further investigation.

Each of the five clusters, labeled 0-4, is characterized by dominant venues as follows:

Cluster Label

Member

Common Venue

0

13

Bakeries

1

3

Mexican and Wine Shops

2

13

Italian and Wine Shops

3

92

Coffee/Cafe, Pizza, Turkish food

4

14

Doner restaurants and Wine Shops

A considerable number of coffee shops and bakeries as well as Turkish food and wine shops are present. Categories indicating "healthy" food such as "vegan / vegetarian places" are not awarded to be unter the Top 5. In fact, such places are very very rare and therefore, it is recommended that stakeholders look into opportunities allover Berlin stations to start a business with organic food and beverages.

3.1 Conclusion

Stakeholders searching for opportunities to open organic food and beverages (incl. vegan / vegetarian dishes) may want to consider setting up their business someplace where competitions are not severe. This study has shown that in the very close proximity of metro/tram stations of Berlin (radius of 100 m) such places don't exist and, therefore, such places are among the best candidates for organic food and beverages location.

3.2 References

[1] https://www.cnb-online.de/hintergruende/zahlen-und-fakten-zum-oepnv/ [2] https://www.oekolandbau.de/handel/marktinformationen/europaeischer-bio-markt-waechst-auf-ueber-40-milliarden-euro/ [3] https://de.wikipedia.org/wiki/Liste_der_Berliner_U-Bahnh%C3%B6fe [4] https://developer.foursquare.com/

[ ]:

[ ]: